

Computer Architectures: Introduction

Todor Stoyanov

Department of Computing and Software
McMaster University

07.09.2022





■ What is this class about?

- Go to www.menti.com and enter number **1771 5615**
- Or scan below



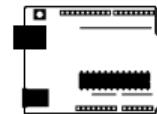
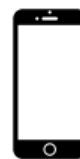
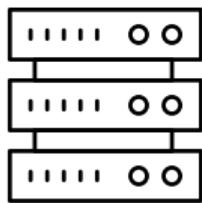
Welcome to 2GA3

- What is this class about?
- Go to www.menti.com and enter number **1771 5615**
- Or scan below





- Why should we care about architecture?





Computer Architectures

■ When should we care about architecture?

- High-performance computation
- Operating systems
- Drivers and hardware acceleration
- Low-level and embedded programming



Computer Architectures

■ When should we care about architecture?

- High-performance computation
- Operating systems
- Drivers and hardware acceleration
- Low-level and embedded programming



Computer Architectures

■ When should we care about architecture?

- High-performance computation
- Operating systems
- Drivers and hardware acceleration
- Low-level and embedded programming



Computer Architectures

■ When should we care about architecture?

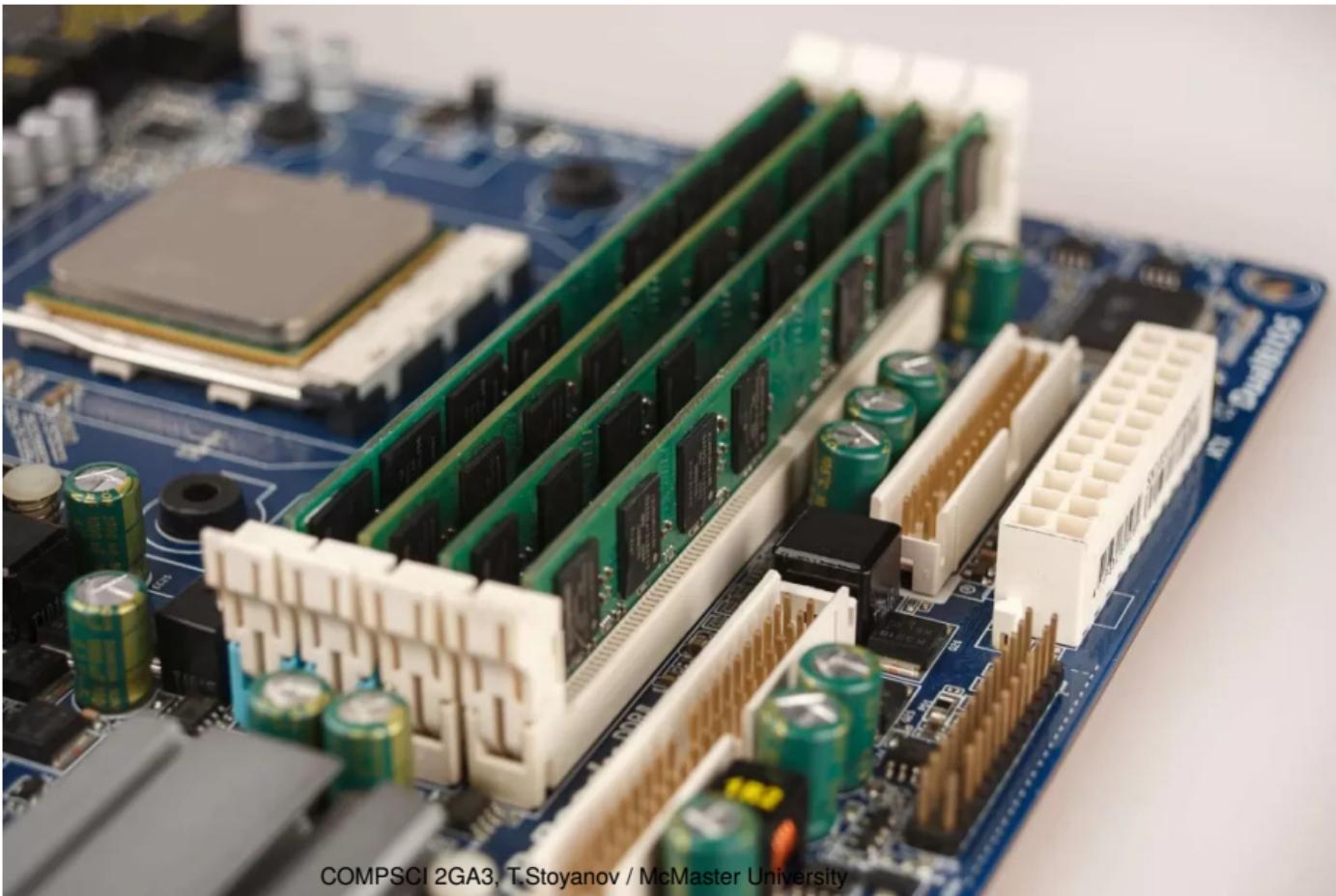
- High-performance computation
- Operating systems
- Drivers and hardware acceleration
- Low-level and embedded programming



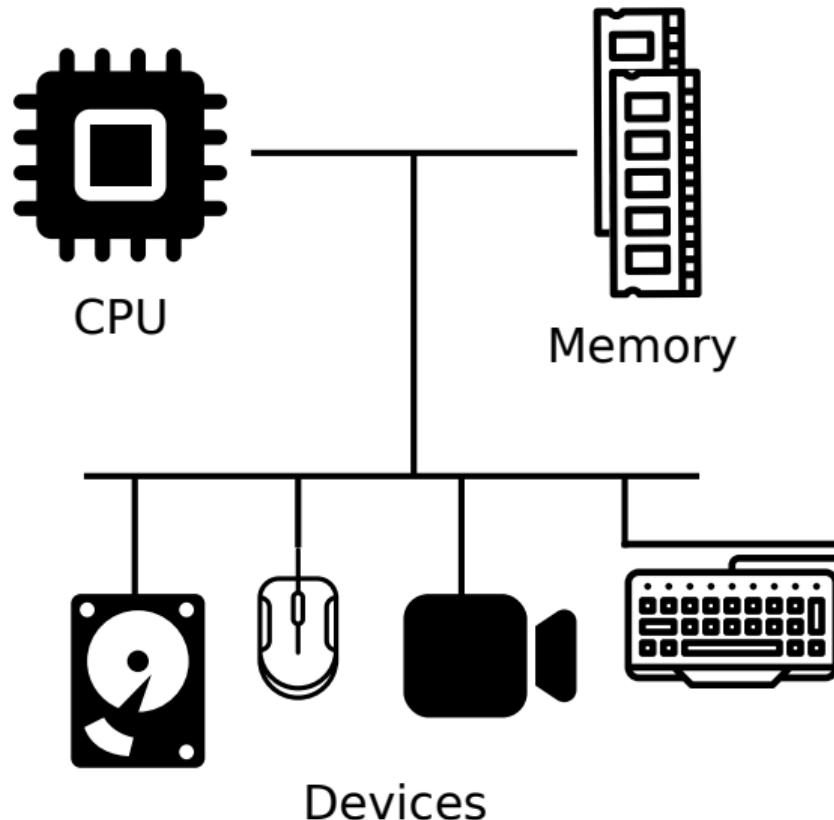
Computer Architectures

- When should we care about architecture?
 - High-performance computation
 - Operating systems
 - Drivers and hardware acceleration
 - Low-level and embedded programming

Components of a Computer



Components of a Computer



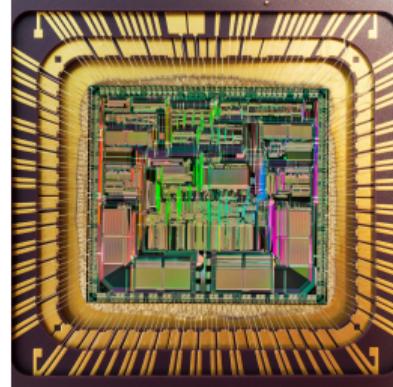


Architectural Design Principles

- How we combine components has implications on performance
- Task-dependent
- Core ideas of computer architecture and hardware design unchanged

Abstraction

- Abstraction in hardware design as key as it is in software
- Hides complexity
- Allows decisions and designs at different levels of detail



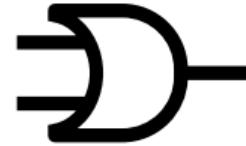
Abstraction

- Abstraction in hardware design as key as it is in software
- Hides complexity
- Allows decisions and designs at different levels of detail



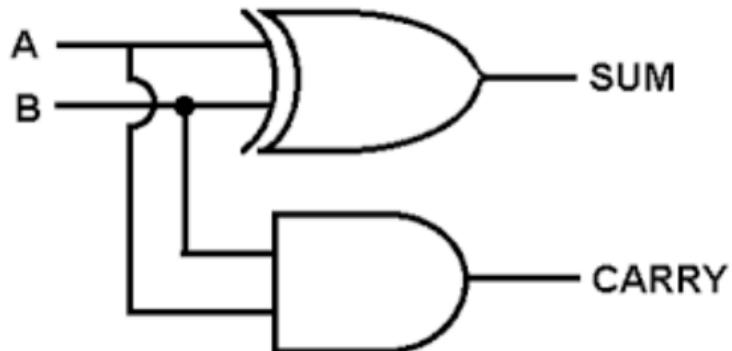
Abstraction

- Abstraction in hardware design as key
as it is in software
- Hides complexity
- Allows decisions and designs at
different levels of detail



Abstraction

- Abstraction in hardware design as key as it is in software
- Hides complexity
- Allows decisions and designs at different levels of detail



Abstraction

- Abstraction in hardware design as key as it is in software
- Hides complexity
- Allows decisions and designs at different levels of detail

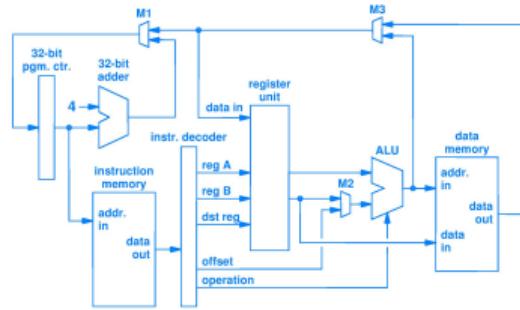


Figure 6.9 Illustration of data paths including data memory.

Redundancy and Replication



Schaerbeek

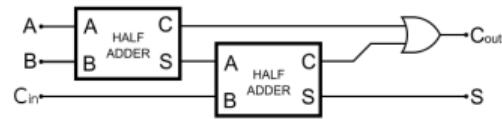
PS: 18,065	CD&U
MR: 17,124	CDF:
ECOLO: 6,177	AGAL
CDH: 5,717	UIUAI
MARIA: 4,510	N-VA
ULAAAMS BLOK: 4,497	FNU
FN: 2,075	FNB:
ULD: 1,921	RWF-I
sp. a-spirit: 1,494	BELG
PCP: 1,398	LIB.
PS: 18,065	CDF: 798
MR: 17,124	AGALEU: 595
ECOLO: 6,177	UIVANT : 563
CDH: 5,717	MARIA: 514
ULAAAMS BLOK: 4,497	N-VA: 333
FN: 2,075	FPC: 308
ULD: 1,921	FNB: 36
sp. a-spirit: 1,494	RWF-RBF: 234
PCP: 1,390	BELG. UNIE-BUB: 223
CD&U: 981	LIB. APPEL: 147

-4096

So what went wrong?

Redundancy and Replication

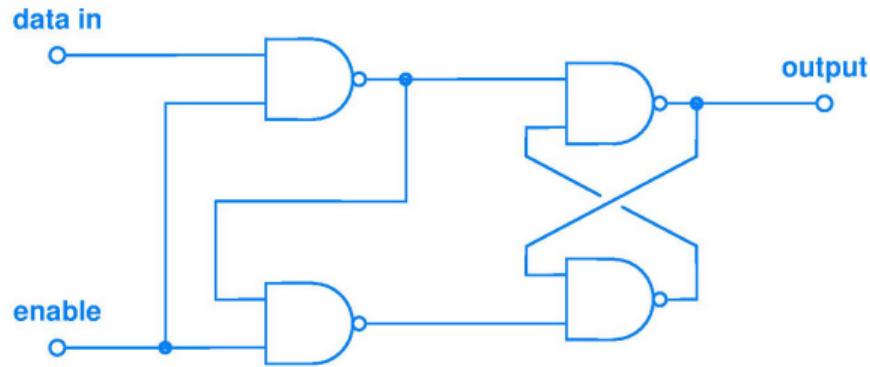
- Hardware has to *always* work as designed
- Formal verification of circuits
- Replication vs Iteration



Parallelism



- If an idea works once ...
 - ... it can work 2^N times as well.
 - Parallelism on a micro- and a macro-level.



Parallelism

- If an idea works once ...
- ... it can work 2^N times as well.
- Parallelism on a micro- and a macro-level.

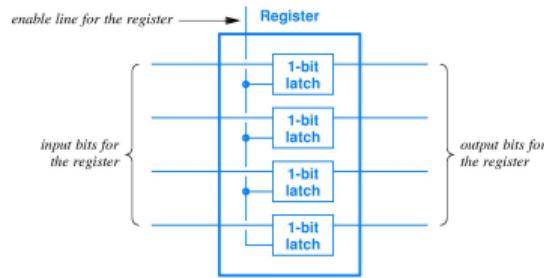
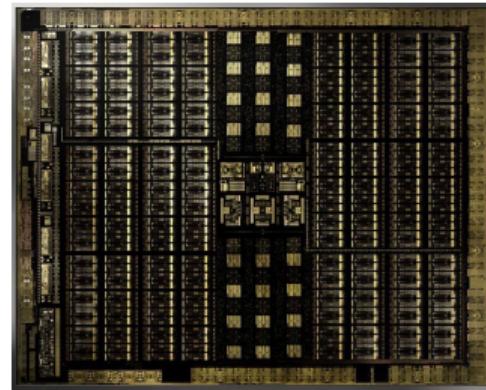


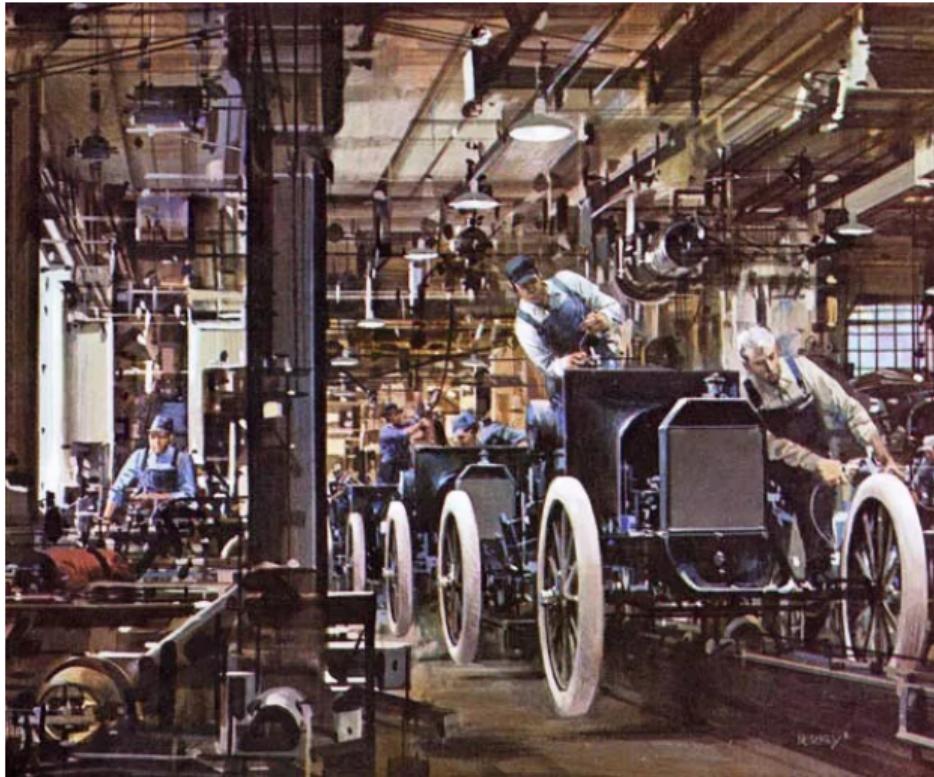
Figure 2.16 A 4-bit register formed from four 1-bit latches.

Parallelism

- If an idea works once ...
- ... it can work 2^N times as well.
- Parallelism on a micro- and a macro-level.



Pipelining





Moore's Law

- Get ahead of the curve so your design is not obsolete.

What will we learn this semester?



Topics we will cover:

- Digital logic basics
- Representing numbers and data
 - Instruction Sets
 - Pipelined processors
 - Microcode and Macrocode
 - Low-level programming languages
- Memory hierarchies
- Caching
- Virtual Memory
- Polling and Interrupts
- Device drivers and I/O

What will we learn this semester?



Topics we will cover:

- Digital logic basics
- Representing numbers and data
- Instruction Sets
- Pipelined processors
- Microcode and Macrocode
- Low-level programming languages
- Memory hierarchies
- Caching
- Virtual Memory
- Polling and Interrupts
- Device drivers and I/O

What will we learn this semester?



Topics we will cover:

- Digital logic basics
- Representing numbers and data
- Instruction Sets
- Pipelined processors
- Microcode and Macrocode
- Low-level programming languages
- Memory hierarchies
- Caching
- Virtual Memory
- Polling and Interrupts
- Device drivers and I/O

What will we learn this semester?



Topics we will cover:

- Digital logic basics
- Representing numbers and data
- Instruction Sets
- Pipelined processors
- Microcode and Macrocode
- Low-level programming languages
- Memory hierarchies
- Caching
- Virtual Memory
- Polling and Interrupts
- Device drivers and I/O



Knowledge and understanding of the following concepts:

- Basis of computation in digital logic;
- Representation of integer and floating point numbers;
- Interaction between processor, memory and I/O;
- Instruction representation and Instruction Set Architecture;
- Interaction between high- and low- level programming languages;
- The functions and implementation of pipelining and parallelism in the CPU;
- Virtual memory and caching;
- Programmed and interrupt-driven I/O.



ILOs

Skills and abilities to:

- Convert between different number representations and reason about representation standards;
- Program simple routines in ARM assembly;
- Interleave the use of assembly and C in a practical programming task;
- Interpret and reason about Instruction Set Architectures;
- Discuss and evaluate the design and performance of ISAs, memory hierarchies, and I/O systems;
- Design and evaluate architecture choices.



Administration

Grading

20% Homework Assignments [5 assignments]

30% Midterm Exam [November 3, TBC]

50% Final Exam

Tutorials

Monday 11:30—12:20 in ETB 238

Monday 09:30—10:20 in JHE A102

Monday 09:30—10:20 in JHE A101

Tuesday 12:30—13:20 in JHE A101

Note: Tutorial instructors may rotate due to conflicting schedules.



Administration

Grading

20% Homework Assignments [5 assignments]

30% Midterm Exam [November 3, TBC]

50% Final Exam

Tutorials

Monday 11:30—12:20 in ETB 238

Monday 09:30—10:20 in JHE A102

Monday 09:30—10:20 in JHE A101

Tuesday 12:30—13:20 in JHE A101

Note: Tutorial instructors may rotate due to conflicting schedules.



Administration

- Academic integrity and other notes: check syllabus
- Health and safety: advised to wear a mask
- Echo360 recordings
- Online lectures, face-to-face tutorials
- Server access for homeworks



Meet the team



Todor Stoyanov



Bhuvesh Chopra



Manish Rawat



Angela Wang



Qi Zhao



A little about you

- Go back to www.menti.com and enter number **1771 5615**
- Or scan below

